# OPEN SOURCE OCR WORKFLOWS FOR HISTORICAL RESEARCH

TOBIAS BLANKE*

*Centre for e-Research, King's College London,*
*26-29 Drury Lane*
*London WC2B 5RL*
*tobias.blanke@kcl.ac.uk*

MICHAEL BRYANT

*Centre for e-Research, King's College London,*
*26-29 Drury Lane*
*London WC2B 5RL*
*michael.bryant@kcl.ac.uk*

MARK HEDGES

*Centre for e-Research, King's College London,*
*26-29 Drury Lane*
*London WC2B 5RL*
*mark.hedges@kcl.ac.uk*

## Abstract

This paper presents an evaluation of open source OCR for supporting research on material in small- to medium-scale historical archives. Our approach was to develop a workflow engine to support the easy customization of the OCR process towards the historical materials. Commercial OCR often fails to deliver sufficient results here, as their processing is optimized towards large-scale commercially-relevant collections. Our paper demonstrates that such a workflow approach allows users to combine commercial engines' ability to read a wider range of character sets with the flexibility of open source tools in terms of customisable pre-processing and layout analysis. We present two of our case studies, which demonstrate how this can be achieved and how OCR can be embedded into wider digitally-enabled historical research. The first case study produces high-quality research-oriented digitisation outputs, utilizing services that we developed to allow for the direct linkage of digitisation image and OCR output. The second case study demonstrates what becomes possible if OCR can be customised directly within a larger research infrastructure for history. In such a scenario, further semantics can be added easily to the workflow, enhancing the research browse experience significantly.

**Keywords** Digital History; OCR; open source; workflows; historical collections.

**Paper type**:
**General review** **Research paper** **Technical paper** **Conceptual paper** **Viewpoint**

## 1. Introduction – Digitisation for Humanities Research

It has been recognised (Jankowski, 2009) that we are at the beginning of a fundamental shift for humanities research and for history in particular. The past decade has seen an unprecedented growth in the quantity of digital material that has become available for humanistic research. This is the result of two complementary developments. First of all, the quantity of *born-digital* material is growing at an ever faster rate, as for all recent historical developments a significant amount of such material has been produced as a kind of live commentary on the events. The story of a supercomputer using sentiment analysis to cover and analyse political revolutions, during the Arab Spring and elsewhere, was widely reported (BBC, 2011).

Even larger, at least in the short term, is the potential for rediscovering the vast amount of historical material in archives through large-scale digitisation efforts. The Google Books project is the most well-known example to date, but the European Union has also invested a large amount of resources to make available online all European cultural heritage. For example, Europeana will soon connect to 20 million digital objects (Ayris, 2010). Such efforts have led to a renewed interest in technologies that would facilitate discovery and analysis of these online digital resources, especially to make use of them in new humanistic research.

Major memory institutions are systematically digitising the material for which they are responsible, but nevertheless digitisation specifically for humanities research is a somewhat piecemeal affair, and is carried out to different extents (e.g. image only or image plus optical character recognition (OCR)) and quality levels. There are key differences between large-scale digitisation efforts (as attempted by Google or in the context of the Europeana initiative) and those that have concentrated more on historical material with a specific focus on humanities research. Firstly, there is the question of resources. Most research funding is project-based and notoriously limited, making it less likely that additional resources are available for buying in OCR expertise. More importantly, however, the material produced by humanities digitisation projects is known to be the result of interpretation. This means that there may be a need to revisit aspects of the digitisation process from time to time when new discourses about the source material emerge. Furthermore, the digitisation needs to be done by staff experienced in research, as otherwise important details might be left out. Digitisation and the creation of digital surrogates are not neutral processes; at least, they need to be considered to be non-static and not finished once the initial digitisation has been done.

Digitisation objects need to reflect the specific research interests that trigger their digitisation if they are to become scholarly research objects. A frequently-quoted example of a large-scale digitisation effort that falls short of scholarly standards is Google's attempt to scan and make available Tristram Shandy (Duguid, 2007). Here, whole pages were left out because they were considered to be misprints, although they were part of the original composition of the novel. Google did not consider it necessary to ask for more input from researchers and scholars in the field.

Current digitisation technology does not reflect such specific research interests in historical document collections very well. In particular, optical character recognition

(OCR) often fails to deliver adequate results. Most commercial OCR software products are proprietary 'black boxes', which provide digitisation staff with little scope for understanding their behaviour or for customising the parameters under which they operate. OCR software manufacturers show a marked reluctance to allow access to their source code, even in a collaborative environment. This use of 'black box' tools, and even more so the outsourcing of OCR processing, leads to a skills and knowledge gap among researchers and archives staff involved in digitisation, which results in a failure to appreciate the problems and opportunities that OCR approaches offer the scholarly community.

Repeatedly identified as critical for access to scholarly and cultural heritage resources (Crane, 2002), developments in OCR can be traced back to the early days of computing. Large-scale automated OCR efforts have always been a dream for heritage libraries and archives (Smith and Merali, 1985), but only with the arrival of new scanner and circuit technologies did OCR applications became more commonplace in public institutions and businesses. New standards for print fonts and paper helped advance the accuracy of OCR further from the 1970s. The next great frontier (Rice et al., 1999) was the machine recognition of handwriting, such as on cheques, for which completely new tools and methodologies had to be developed. Later on, computers became more powerful, and evaluation conferences such as IWFHR (International Workshop on Frontiers in Handwriting Recognition) and CDAR (International Conference on Document Analysis and Recognition) helped to develop OCR algorithms further (Mori et al., 1999). Nowadays, handwriting recognition has reached a mass-market with tablet PCs, and is used in everyday environments, yet it remains among the most difficult tasks in OCR (Thompson, July 1997).

However, all these development were linked mainly to well-defined commercial use cases for business environments. Historical documents, with their immanent constraints of low quality and unusual character sets, remain a challenge to OCR. Especially in the context of historical documents, OCR will almost never produce 100% accuracy and an exact digital surrogate of the original (Haigh, November 1996). Therefore, the challenges to full automation of the recognition of historical texts seem insurmountable, as Brocks et al. (2001) discuss.

Flexible methodologies need to be developed to optimise OCR processes for historical documents as far as is possible. Recently, there has been renewed research interest in the field, with new methodologies (Vamvakas et al., 2008) and attempts to build production-level services for European cultural heritage institutions (Ploeger, 2009). This paper contributes to these developments by presenting a general-purpose OCR environment for historical documents that is based around open source technologies, which are easily customisable. Volk et al. (2011) also experimented with open source technologies to address the specific challenges of historical documents, but concluded that their recognition rate is too low, which is why they concentrated on commercial products. In (Blanke et al., 2011), we have shown that many open source products are by now mature and deliver acceptable recognition rates for historical documents, though they are still behind commercial tools when it comes to stable character recognition. This is why we argue in this paper that the best results will be achieved by combining the flexibility of open source tools with the mature character models of commercial products.

These thoughts are the beginning of our reinterpretation of the process of OCR for historical archives as a process that is not simply neutral but needs to be understood in the context of the humanities research process, which implies that researchers and their support staff need to be given the opportunity to customise the process. In this work, we attempt to integrate OCR seamlessly into workflows for the humanities and to embed a full portfolio of digitisation-related skills within expert historical archives. In our opinion this cannot be achieved with commercial software alone, and our work has focussed largely on open source technologies for OCR.

In order to embed OCR in institutional workflows of historical archives and increase the understanding of the steps involved, we developed a workflow system for OCR tools for use by small- and medium-sized digital archives, focussing principally on tools that were open source. Initially, we expected the outcome to be a set of recommendations and best practices delivered in the form of documentation. However, our work ultimately led to the development of software tools that we felt provided a better platform than documentation alone for demonstrating our workflow ideas, so eventually more effort was dedicated to the development of the software. This paper presents our workflow approach and evaluates its use in small-scale historical archives.

In an attempt to explore potential solutions for open source OCR workflows we developed an application called OCRopodium Web Processing (OWP). OCRopodium was the name of the JISC-funded project under which this work was carried out, and its aim was to investigate open source OCR technologies and their use in historical archives to support data-driven humanities research. Built on a number of open source technologies, OWP provides an abstraction layer on top of open source OCR tools, along with a web-based front-end for managing a full OCR workflow in a user-friendly manner.

OWP is intended to be a lightweight environment for developing *procedural* OCR workflows built using off-the-shelf tools. It is inspired by advanced procedural editing environments used in audio, video, and image processing, as well as full-featured scientific workflow tools like Taverna (Oinn et al., 2006), with the additional goals of being simple as well as flexible for a client-server environment with a web-based front-end.

In Section 2 we introduce the OCR workflow from the perspective of open source tools and the requirements for historical archives. Section 3 presents our OCR workflow environment. We describe the technologies and methodologies used to develop the environment as well as how we embed it into the institutional infrastructures at King's College London. In Section 4, we detail how researchers and digitisation staff can interact directly with the various stages of the OCR process within our environment. We evaluated our solutions with various case studies, two of which we present in Section 5. The first one illustrates a number of generic challenges that we have encountered frequently in research-oriented digitisation projects, while the second one examines how our OCR workflows can be included in larger-scale digital research for history.

## 2. Open source OCR workflow for historical archives

This section describes the experimental setup for developing scientific workflows for historical research in which OCR is an integral part. Section 2.1 presents the OCR tools that we used for our experiments and case studies. We have investigated many more OCR tools and services; for a complete overview compare (Blanke et al., 2011). Section 2.2 explains the OCR process, decomposes it into some of its discrete steps, and

describes the relevance of these steps to historical archives. In this way we demonstrate that it is possible to split up the process into discrete steps that can be controlled by research archive staff.

### 2.1. *OCR Tools used*

We analysed a variety of open source OCR tools for processing documents in historical archives. For our experiments, we used the relatively mature Tesseract 3.0 recognition engine as well as the research-focused and continually-evolving OCRopus toolkit (Breuel, 2009). OCRopus, a set of OCR tools developed by the Image Understanding and Pattern Recognition group (IUPR) at the University of Kaiserslautern, stood out as being the most accessible in terms of facilitating active modification by third-parties and was thus the focus of much of our efforts. It is comprised of a large number of discrete components, which is intended to make adding missing functionality or customising behaviour a more straightforward task.

Tesseract (Smith, 2007) was originally developed by Hewlett Packard from the mid-1980s, but saw little to no development until the code was released under an open source license in 2005. It is currently being worked on at Google, with the latest version 3 being released in late 2010. Up to version 2, Tesseract was more of a "pure" character-recognition tool, since it contained no facilities for handling non-binary (colour or grey-scale) images or page layout analysis. Fairly simple implementations of these features have been added in version 3 via the integration of the third-party Leptonica[†] library, which improves Tesseract's general-purpose usefulness.

For our benchmarking, we also included commercial products that ran on Linux and featured similar command-line interfaces (CLI) to the various open source offerings. We mainly used Finereader 8.0 CLI[‡], which is a pre-packaged Linux version of Abbyy's developer-focussed (and relatively expensive) OCR engine development kit. Several page-limited licenses are available, with the cheapest offering 12,000 pages on a single computer for £100. Finereader 8.0 CLI provides a good range of output formats and a limited range of options for tuning its engine to different material.

Next, the OCR process is broken down into discrete steps, and the challenges these pose for a full procedural workflow in historical archives are explained. In Section 3, we pick up these steps and support them individually in our workflow environment.

### 2.2. *Requirements for the individual steps*

An OCR workflow can be broadly described as consisting of four principal stages: pre-processing, layout analysis, recognition and correction (Bryant et al., 2010) (Mori et al., 1992). In this section we will briefly relate these steps to the specific challenges in historical archives.

#### *2.2.1 Pre-processing*

Pre-processing is the act of rendering a raw page scan into a state where the OCR recognition engine is able to achieve optimal accuracy rates. Often termed simply "binarisation", the pre-processing stage usually consists of several mostly discrete sub-steps. In general terms, the process contains at least the following steps:

[†] http://www.leptonica.com/
[‡] http://www.abbyy.com

- Binarisation itself describes the process of converting the colour or grey-scale scan into a binary format, isolating as much of the text as possible. There are many different algorithms for performing image binarisation, and it may or may not take place after initial de-noising of the scan.
- De-noising is the process of removing undesirable artefacts from an image to prevent them being confused for textual content. Typical "noise" found in historical archives could take the form of grain on the paper surface, bleed-through from the back sheet, or other stains and dirt present on the source material.
- De-skewing/warping means heuristically aligning the image so that lines of text are orientated as near to the horizontal plane as possible. Most scans will have some degree of page skew, and effective de-skewing is vital to ensure the maximum effectiveness of page segmentation algorithms and character recognition.

Less common pre-processing steps, not appropriate on all types of material, include:

- Half-tone removal or converting half-tone images (i.e. those where dense patterns of dots form the impression of solid blocks) to contiguous solid regions.
- Heuristically removing non-text elements, such as images and line-rules, from a page of text.

Most consumer OCR engines automate the recognition process to a very high degree, providing few opportunities for the skilled user to tailor the process for specific types of material. At the other end of the scale, "raw" engines such as Tesseract 2.0 expect their input in an already optimised, binary form. OCRopus is almost unique in providing a suite of pre-processing tools that are largely decoupled from its segmentation or text-recognition components. This is the reason why we have chosen to concentrate on OCRopus but complement it with other tools for steps where these show superior performance.

Although the automated approach to pre-processing taken by most commercial OCR products is appropriate for the general end-user, historical material can have much more demanding characteristics, such as higher noise, lower contrast and unusual colourisation. To deal with these issues it is necessary to have better control over the process than typically possible with "black-box" software.

In the following section, we discuss two parts of the processing step in more detail: page segmentation and recognition, both key to understanding structure and content of document respectively.

### *2.2.2 Layout Analysis*

Page segmentation is the process of extracting structural elements from an image and interpreting the semantic context of those elements. In most cases, the end product will be a collection of page sections: paragraphs, non-text elements and lines of text in reading order.

The difficulty of performing effective page segmentation varies greatly with the source material. At one end of the scale, a page from an English-language novel would probably consist of a single column of text running from top to bottom. At the other end, a newspaper or journal page might comprise multiple columns, interspersed with figures

and headings. In these cases, even when the position of the lines can be correctly determined, figuring out the correct reading order of those lines is fraught with complexities. Even with simple single-block page layouts, determining the locations of the individual lines in a block can present serious challenges, especially if the block in question is warped, skewed or contains characters with overlapping descenders and ascenders, all of which is common for historical documents.

### *2.2.3 Recognition*

For most OCR engines, the primary unit of recognition is a single line of text, so that the recognition stage may be thought of as applying the result of the page layout analysis to the pre-processed binary image, attempting to transcribe the input one line at a time. The OCR engine then attempts to break the line into individual characters. Since characters in a typical historical text are frequently broken or overlapping, white space separation is of limited use in assisting this process, which is typically achieved instead using a complex interaction of character and language modelling (Blanke et al., 2011). Since there is a large probabilistic element to the output results, some OCR engines can output character confidences, which indicate how likely the text is to be correct. These confidence scores can be of great value when the output material undergoes post-processing and correction later on.

### *2.2.4. Post-processing and Correction*

Correction is perhaps the most important stage in the OCR workflow for historical archives, as well as the most time-consuming. This has led the European IMPACT project to focus on crowd-sourcing technologies to support the correction process (Neudecker and Tzadok, 2010). IMPACT also covers historical OCR, but is mainly using commercial OCR engines and developing tools that appear more suited to large institutions such as national libraries.

Inevitably, the correction stage involves mostly manual data entry, and is thus outside the scope of automated workflow tools. There is, however, a certain amount of post-processing that can sometimes be used to correct common OCR transcription errors. Thus, we had to develop our own interactive post-processing interface, which we present in Section 4.

This concludes our discussion of the steps involved in the OCR process and their relevance for historical archives. The following sections present our solutions for the challenges involved in each of these steps. We begin by developing our workflow environment in Section 3. This environment allows for the chaining of OCR tools, before in Section 4 manual interaction to optimise the automated processing is discussed. We found that only the tight integration of both automation and interaction delivers the best recognition results.

## 3. Processing historical documents: The OWP environment

This section details our OWP workflow environment. Section 3.1 presents the open source technologies in order to chain OCR tools, while Section 3.2 investigates how OWP can be used to process historical collections. Section 3.4, finally, demonstrates how we can embed OWP to optimise the work on historical collections within an institutional

environment by demonstrating how it interacts with an institutional repository and a batch-processing environment that can make use of Cloud-based resources (Chen et al., 2010).

### 3.1. *The OWP Workflow environment*

The OWP workflow application is based on the principles of visual programming environments (Cox and Gauvin, 2011). It allows the user to build custom workflows comprised of discrete processes. The workflows need not be purely linear; instead they take the form of a graph, specifically a directed acyclic graph (DAG). The DAG is comprised of connected nodes, which each perform a discrete function. Nodes can be thought of as functions which take one or more inputs and evaluate to produce a result. Obtaining a result from the entire DAG, therefore, involves finding the root node and asking it to evaluate its result. To do this, it must ask its input nodes to evaluate, which must each likewise evaluate *their* inputs. The chain of evaluation continues *up* the DAG from root to tip, and data can be seen as flowing *down*, culminating in the root node's output.

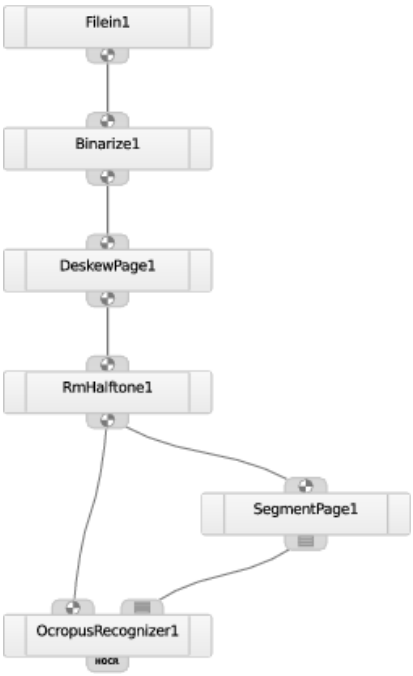



**Figure 1: A simple workflow**

This scheme can be best illustrated by a simple example such as Figure 1. In this case the root node, labelled "OcropusRecognizer1" is connected to two inputs, "RmHalftone1" and "SegmentPage1"[§]. In order for OcropusRecognizer1 to produce a result, it must obtain the result of its two inputs. Each input then does the same for *its own* inputs, until

[§] The names are derived from the types of node they represent, "RmHalftone**1**" being the first "RmHalftone" node in the tree.

the chain of evaluation reaches the "filein1" node, which simply produces its result by reading a file from disk.

The data, therefore, flows in the opposite way to the order of evaluation. In Figure 1 results are produced in order to:

1. Read an image file
2. Binarise the image
3. Deskew the image
4. Remove half tone
5. Segment the image
6. Recognise the text

This procedural scheme has some significant advantages, since it lets the research user control and view how each step of the process contributes towards the final result (in this case some OCR'd text). If we, for example, wish to see whether the "Deskew" node is improving the OCR text output, we can simply toggle it on or off via the GUI whilst viewing the output of the recogniser node. Likewise, it makes it possible to compare nodes that perform the same function in different ways - e.g. two page segmentation nodes that use different algorithms - by linking them in parallel and toggling which one is active. We have tested it in various case studies (see Section 5) and users have generally found the environment easy to use.

The node graph structure is serialised to the JSON data format. For convenience the OWP stores presets in its local database, but they are not in any way specific to the web application and can be easily downloaded, shared or stored in a digital repository. Executing procedural OCR presets is likewise not tied specifically to the OWP web application: downloaded presets can be run locally using a simple command-line tool.

### 3.2. *Processing Steps*

As outlined above, pre-processing with OWP is a case of chaining together processing steps that are appropriate for the source image. The source file is read as a greyscale image, after which filters can be applied to perform tasks such as de-noising and rotation. The greyscale image is afterwards binarised using one of several available methods, most of which use components from the OCRopus tool set. Additional filters would then be applied to the binary image to prepare it for OCR. Binary filters would typically be more specific to the requirements of the OCR engine, performing functions such as de-skewing, edge-removal and removal of non-text components.

At the time of writing, the OCR-specific binary pre-processing tools integrated into the OWP come from the OCRopus tool-kit. There are also several filters derived from the Python Imaging Library (PIL) for more general-purpose image processing functions (e.g. rotation, crop and sharpen), which we have interfaced seamlessly with the OCRopus tools.

We have tried to build on the existing page-segmentation features of OCRopus in ways that make sense for historical OCR. As in pre-processing, this does in some cases mean exposing those features in ways that allow greater customisation on the part of the end-user. Whilst the ideal OCR system should be able to correctly analyse a wide variety of page formats without user intervention, the variable quality and more frequent artefacts found in historical material make this more challenging.

Given that it is common in historical collections that groups of individual pages will not conform to a standard layout, it is a pragmatic to require more input from the user at the outset in order to ensure consistent treatment. This approach is best illustrated by taking one of our test collections as an example. The Stormont Papers corpus (further discussed in case study 1 in Section 5.1) consists of four hundred images, all of which have the same page layout: a single date-line at the top and two columns (Figure 2).

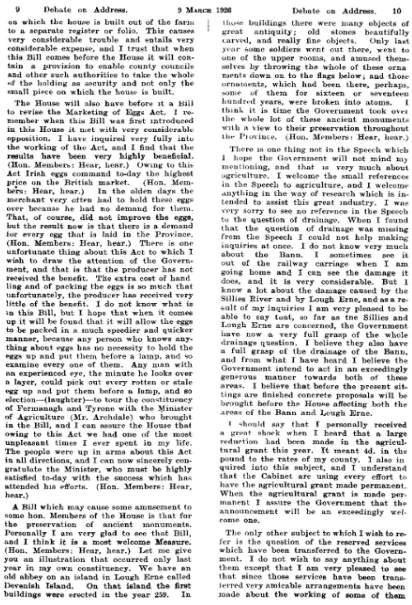
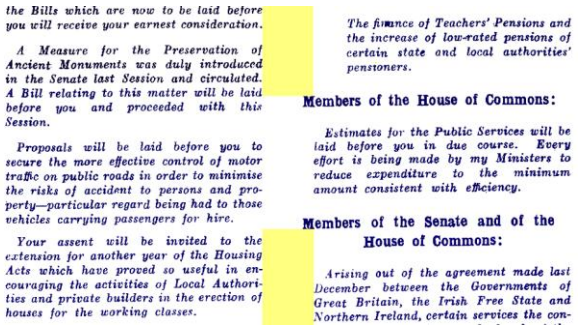


**Figure 2: Example Stormont Papers two-column-plus-date-line layout**

We have found that, though font spacing is very variable and column skew is quite high, OCRopus performs quite well using its standard page segmentation component called SegmentPageByRAST (Breuel, 1992). Nevertheless, some problems seem to reoccur. For instance, the header line might not be discerned as a distinct from the two following columns. Sometimes, more difficult-to-anticipate issues crop up, such as that highlighted in Figure 3, where the pronounced leftward outdent of the headings has misled the layout analysis algorithm into hypothesising a more complex layout than is actually present on the page.



*the Bills which are now to be laid before you will receive your earnest consideration.*

*A Measure for the Preservation of Ancient Monuments was duly introduced in the Senate last Session and circulated. A Bill relating to this matter will be laid before you and proceeded with this Session.*

*Proposals will be laid before you to secure the more effective control of motor traffic on public roads in order to minimise the risks of accident to persons and property—particular regard being had to those vehicles carrying passengers for hire.*

*Your assent will be invited to the extension for another year of the Housing Acts which have proved so useful in encouraging the activities of Local Authorities and private builders in the erection of houses for the working classes.*

*The finance of Teachers' Pensions and the increase of low-rated pensions of certain state and local authorities' pensioners.*

**Members of the House of Commons:**

*Estimates for the Public Services will be laid before you in due course. Every effort is being made by my Ministers to reduce expenditure to the minimum amount consistent with efficiency.*

**Members of the Senate and of the House of Commons:**

*Arising out of the agreement made last December between the Governments of Great Britain, the Irish Free State and Northern Ireland, certain services the con-*

**Figure 3: Example layout analysis error.**

When dealing with material containing typographic variability but quite uniform layouts, a page-segmentation algorithm that is capable of understanding very complex layouts can sometimes work against achieving consistent outcomes.

### 3.3. *Guided evaluation of workflows*

In addition to nodes which perform transformations on input images or text, we have also integrated into the OWP workflow environment tools which aim to help the user arrive at the best possible workflow for a given set of documents. In a typical scenario the user would select a small number of representative images from a much larger set of documents requiring OCR, and obtain (using a non-specialised workflow) the transcript text for each of them, which is subsequently corrected. This transcript text can then be used as a "ground-truth" for interactively evaluating the efficacy of specialised workflows for use on the corpus as a whole, enabling higher accuracy to be obtained and requiring less eventual correction.

The basis of this evaluation workflow is the set of command-line tools developed by the University of Nevada Las Vegas (UNLV) Information Science Research Institute (ISRI) for use on their annual OCR accuracy tests between 1992 and 1996. The UNLV-ISRI tools compare the OCR output with a ground-truth text and provide reports detailing key metrics such as character and word accuracy, and shifts in text-ordering that indicate the presence of errors resulting from possible mis-segmentation of the input image. These tools have been integrated as post-processing nodes within the OWP environment.

Connecting the UNLV-ISRI evaluation tools to the output of an OCR workflow and comparing it to a ground-truth text provides the user with the means to interactively determine exactly how each step of workflow contributes (positively or negatively) to the final accuracy score. For a large document set, this can yield valuable insights into how and where errors are occurring, and provides a more objective basis for developing workflows than relying on the human eye alone. Moreover, since a single percentage-point of additional accuracy translates to a large amount of correction that does not then have to be done, the up-front effort of producing sample transcripts for use as ground-truth is very often a worthwhile investment.

This summarises some challenges in specifying processing steps for OCRing historical documents. OWP allows archive staff to customise the processing according to their needs. At the same time, OWP interacts seamlessly with institutional archival information systems and with repositories and other local environments.

### 3.4. *Embedding in institutional infrastructures*

We offer two principal means of supporting the processing of documents in a traditional digital archive environment. Firstly, we support direct link-up with the institutional repository system and so that it can be used to store intermediate results directly. This provides much more effective control of outputs for research and archive staff. Secondly, we offer infrastructures for executing the more computing-heavy components of the OCR process, allowing the archive to use either their own local network or to outsource computing to the Cloud.

Our workflows produce outputs in a form suitable for ingest into King's College's Fedora-based digital repository infrastructure (Payette and Lagoze, 1998), and the outputs of the workflow conform to content models used in this infrastructure. For the storage of OCR output text and metadata, we principally employ the HOCR format, an open standard that provides a superset of XHTML with conventions for describing OCR-specific properties (Breuel, 2009). Most open source OCR tools have built-in support for outputting of HOCR data, but for those that do not (and also for commercial tools such as Abbyy Finereader 8.0 CLI) we can utilise XSLT stylesheets to transform application-specific output into vendor-neutral formats. Since HOCR is derived from HTML it uses the same mark-up elements to describe page structure and layout, with the advantage that files are viewable in any standard web browser. The HOCR output is saved in the repository linked to the original digitisation image.

Fedora, however, allows us to not just store objects but to link these with so-called disseminators (Payette and Lagoze, 1998). These are standard services for transforming the underlying objects. For instance, an XML document can be linked to services containing XSLT stylesheets in order to transform it into various standard publication formats such as HTML or PDF. We use Fedora's disseminator infrastructure to store, along with a digitisation image, its optimal recognition workflow, as determined using the OWP environment. This way we can cluster digitisation images in our repository infrastructure with respect to their optimised OCR workflows. Large clusters of related document types can then be processed in optimised batches.

OWP is designed around the notion of processing batches of pages with similar characteristics, as would typically be the case when performing OCR on a more-or-less homogeneous document collection. Once the user has determined that there is sufficient commonality among a set of page scans to treat them as a single batch, they can be marked as related in the Fedora repository and linked to the same OCR workflow by using the Fedora disseminators.

Having determined good pre-processing and segmentation settings, the user can process all related image files in a batch. After launching the batch, the progress of each individual page can be tracked from a separate window (Figure 4), with tasks able to be cancelled or restarted on demand. In testing, batches of up to two-thousand files have been run successfully. The system provides the option of processing the batch using our institutional Cloud, but, in principle, any network could be used for the distributed processing of OCR jobs.

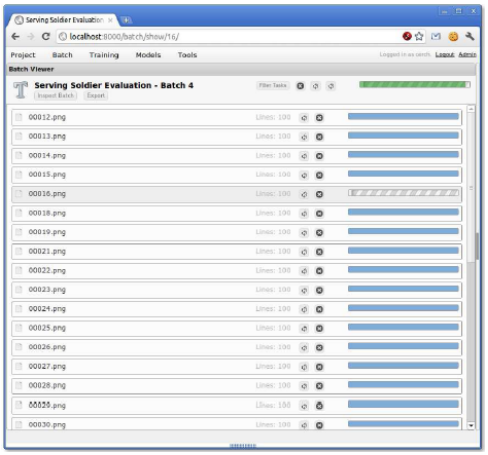



**Figure 4: OWP batch processing view**

Speed, simplicity and scalability were all important initial considerations when mapping out the design of our OCR processing system, but were generally difficult to reconcile. We will discuss this in detail in our first case study in Section 5.1, where we present a typical OCR workflow for historical archives. The second case study from Section 5.2 then concentrates on embedding the OCR service we have developed into larger scientific workflows for historical analysis. Here, we showcase our experiment to extract semantics from historical documents by linking in external information extraction workflows. The node-based design of OWP's procedural workflow makes it a very simple task to employ external web services in addition to (or in place of) native components. Whilst at present there is no way to perform *generic* web service processing, our OWP experiments included proof-of-concept nodes for fetching enrichment and text-analysis data from HOCR output using, for instance, DBPedia's spotlight service (Mendes et al., 2011).

But first, we need to cover the final customisation we have developed. As presented in Section 2.2, for a digitisation project in historical archives, it is essential to have advanced facilities to correct the OCR results. That is why we developed extensive services to support the interaction with source material at all stages of the processing by directly interacting with the repository.

## 4. Interactive Correction GUI

One objective of our OCR application was to provide tools for the complete OCR process, from scanned images to ingestion in a digital repository. One of the most important steps in this process is the manual correction of machine OCR output. Because it is so crucial to the quality of the final output and the most labour-intensive part of the process, designing an effective workflow for OCR correction presents both challenges and significant opportunities to improve on common case scenarios and adjust to the requirements of historical documents.

Users have an existing investment in knowledge and expertise of common word-processing tools, which entails that OCR correction is often performed using common offline word-processing tools like Microsoft Word. Given the labour-intensive nature of

the task, the existing expertise possessed by digitisation staff in using these tools is a very important factor in the efficiency of the overall process. This mitigates against introducing new and different methods for text correction. Furthermore, current workflows rely on the customisability of mainstream word-processors. Usually their macro facilities address persistent misspellings and grammar issues, but may also be used to reformat OCR output.

Nevertheless, a typical, non-specialised correction workflow that utilises tools like Word has major downsides in terms of data integrity, data management, preservation and general efficiency. One of the most pressing issues is the potential loss in digitisation information. During the necessary format conversions metadata is lost. Most OCR engines have the ability to encode formatting information and other important metadata in their raw transcription output, usually in a vendor-specific XML schemas or using standards such as HOCR. It is our experience, however, that this valuable metadata rarely emerges intact from the correction process if standard word processors are used.

In addition to metadata loss, there is an issue with the integration of corrected OCR text into the overall OCR workflow. Use of conventional word-processing software for OCR correction generates an additional set of files containing the corrected results. Whilst many such tools include some form of internal revision control, this is usually highly vendor-specific and not accessible by external tools. If the digitisation process mandates that intermediate revisions are saved separately, as it is the case for high-quality research digitisations, decisions will later have to be made regarding whether or not to preserve this correction data along with the final copy. Either way, the onus is placed on digitisation staff to adopt suitable data management practises to ensure the integrity and safety of these additional files.

Finally, use of a non-integrated solution such as a word-processor for correction of OCR text presents one particular efficiency problem for the user: lack of a context-sensitive visual link between the raw text and the source image. Unlike a touch-typist transcribing a piece of text directly, research and archive staff needs to continually refer between errors in the raw OCR text and the part of the image to which those errors correspond. The need to do this will be particularly acute if a document's language is archaic or unknown to the user, or if the quality of the raw OCR is very bad – both are common features in historical archives. This jumping between contexts, and having to relocate the point in the image to which some given text belongs, is a major constraint on the speed of OCR correction.

Our OWP approach addresses such issues and tries to embed the correction directly into the OCR workflow. Our transcript correction approach tries to alleviate the key problems highlighted above, whilst lowering the barriers to adoption as much as possible. Transcript correction is fully integrated into our processing. Many benefits derive simply from being a web application, namely: platform-independence, access to the centralised project data from multiple machines in multiple locations and no licensing costs. Nonetheless, developing truly interactive and responsive web applications poses its own challenges, and doing so pushes the limits of current web standards.

Ultimately, our key assumption is that while normal word-processing tasks demand maximum flexibility, OCR correction is intrinsically constrained by its source image. This is especially true if we speak of research digitisation where the translation from the image into the text is a question of interpretation. Our correction is geared towards

retaining in the final output as much of the original machine-captured metadata as possible and direct linkage to the underlying source object. Moreover, we take advantage of the positional metadata in structural page elements to provide contextual assistance to the user that can significantly ease the correction process. The general flow is shown in Figure 5.

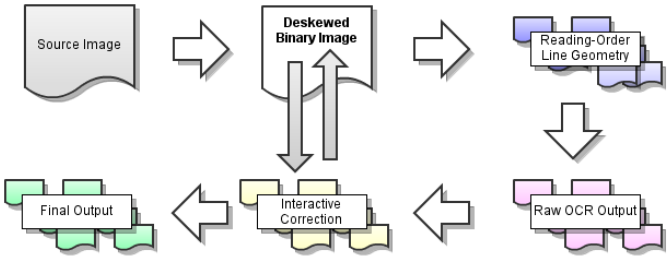



**Figure 5: Interactive correction workflow**

First, the source image is binarised and de-skewed. The binarised page image is then segmented to locate the text lines in *reading order*. Afterwards, automated OCR is performed, before finally the OCR is corrected with reference to the deskewed binary image from step 1.

The interface we present in OWP consists of a two-pane view showing the raw OCR transcript and the binarised image side-by-side. Text in the transcript pane can be viewed in either a single-column mode or positioned as per the source. These layout options make use of positional metadata captured at the OCR stage, and the formatting is otherwise fixed. To preserve the basic page structure of the source image, lines are editable only as discrete units; that is, the editing cursor is constrained within the bounds of the single line.

Constraining the document's editability in this fairly rigid manner certainly imposes limitations on the user's ability to perform free-flowing text correction, but it also allows us to provide a direct link between portions of the transcript and their context within the source image. In our proof-of-concept this link is presented in the form of matching highlights in each pane as visualised in Figure 6.

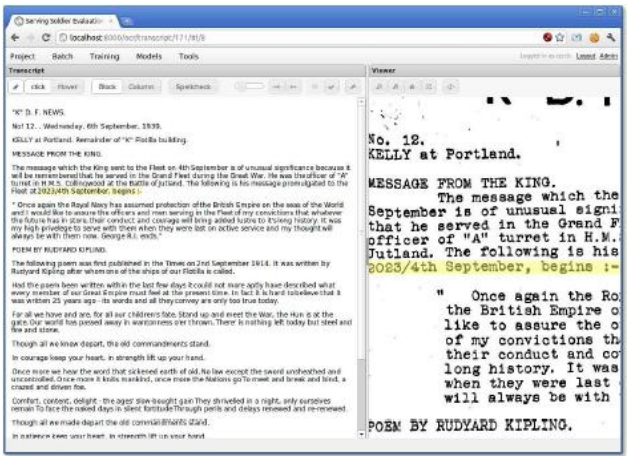



**Figure 6: OWP two-pane transcript correction**

When the user first opens the transcript editing interface, the tab key can be used to navigate through lines in the transcript. The corresponding portion of the binary image is centred and highlighted in the second pane. When a transcript error is spotted, the user can either double-click the line in question, or press the F2-key (as if editing a cell in Microsoft Excel.) The given line is put into edit mode, which allows the user to alter the text in free form fashion until it matches the image. Pressing "Enter" saves the changes, whilst hitting the tab saves the current line and puts the following one into edit mode, allowing the user to navigate through the document in an intuitive manner.

Whilst it is undoubtedly a challenge to incorporate into a bespoke editing system many of the features that users are accustomed to from their experience with mainstream tools, our OWP correction environment – while still basic - does offer multiple undo/redo, as well as interactive spell-checking (based on the GNU Aspell programme) and find/replace tools. Static text-transformation functions – something for which word-processing macros are commonly employed – are best used within our procedural workflow environment, as post-processing nodes chained to the output of the text recogniser.

With the basic aim of taking document management out of the users' hands and integrate it directly in the institutional infrastructure, we have also implemented a repository link that automatically saves incremented versions of an OCR'd page as the user corrects it. The current transcript for a given image is by default the most recently edited version, though it is possible to retrieve and export earlier versions.

This concludes the discussions of our system. The next section presents two of our case studies, where we have evaluated our system with users of typical digital humanities archives.

## 5. Case Studies

The two case studies presented here offer complementary insights into how the environment could be used in practice. The first demonstrates how it can support a stand-alone scholarly digitisation project, in this instance the Stormont Papers, while the second

describes our plans for integrating it as a service within a larger research infrastructure, specifically to support research-oriented digitisation in distributed Holocaust archives.

### 5.1. *Case Study 1: The Stormont Papers*

The Stormont Papers project (Dunning et al., 2007) was a fairly typical medium-sized scholarly digitisation project. The papers offer online access to over 50 years of Parliamentary Debates of the devolved government of Northern Ireland, and comprise over 100,000 printed pages of parliamentary discussion on a wide range of issues. This case study suggests how we can address a number of typical problems that arise when digitising such historical archival material using a combination of automated processing and user interaction. We will first describe the general OCR workflow that we used for the Stormont Papers, and then demonstrate how OWP can be used to chain the most effective tools for a particular task.

Figure 7 shows how one can build OWP workflows interactively. It begins with a fairly standard pre-processing chain consisting of a binarisation node (using OCRopus's Sauvola algorithm (Breuel, 1992)), followed by nodes that run border cleanup, de-skew and large (non-text) element removal operations. The output is then fed into two page-segmentation nodes wired in parallel, whose outputs are recombined with a switch node. The first segmentation uses the described RAST algorithm, while the second uses our own customisation based on user hints, which we describe below. The switch node simply passes through unchanged one of its two inputs, but it provides a convenient way of interactively comparing the output from two functionally equivalent nodes using the OWP GUI. Finally, a Tesseract recognition node is handed both the pre-processed binary image and the layout analysis data to perform the actual text recognition.

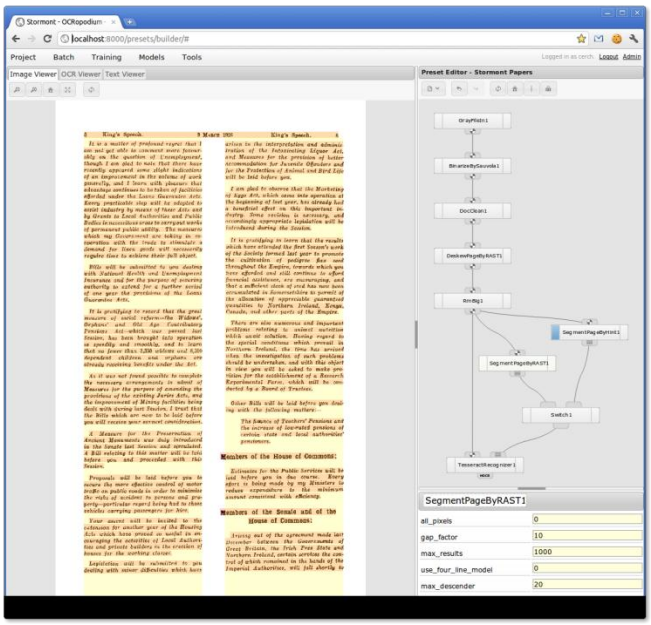



**Figure 7: OWP interactive workflow building interface**

In all our case studies, we employed the various OCRopus components in a native fashion, i.e. making use of its application programming interface (API) directly, while providing scripted wrappers to other standalone command-line tools. This approach abstracts the interface differences between various OCR engines and allows them to be used and compared in an intuitive manner via the OWP preset builder. At present, only the OCRopus and Tesseract engines allow us to integrate bespoke page layout analysis tools, since they provide the means to operate on individual line images; programs such as Abbyy Finereader 8.0 CLI only accept full-page input and are therefore wedded to their own in-built layout-analysis tools.[**]

This describes our general interactive workflow setup. However, as historical OCR raises many specific issues that cannot be addressed by such an automated workflow, it was necessary to integrate further user interaction. An already quoted example is page segmentation; the Stormont material contains a number of pages with idiosyncratic page layouts and we decided to support the research staff by providing dedicated services that make these easier to handle. Problems with the page structure are common when digitising historical archives. In this case, the exceptions occur in a relatively small proportion of the pages, and one solution would be to accept them simply as a given and require the user to intervene when necessary.

Generally speaking, there are two ways to achieve this:

1. The first potential solution to intermittent page segmentation errors is for the user to manually re-order the parts of the transcript that were mis-segmented. This entails various problems: aside from it being a somewhat confusing task to re-arrange out-of-context fragments, this also means that the positional metadata – that is, the information telling us where in the source image a given text-line derived – will usually be lost, or worse, rendered incorrect.
2. An alternative to redoing the page *transcript* is to redo the page segmentation with user input. In this case the user would, using a specialised interactive tool, manually outline the various page components and explicitly provide the correct reading-order. Each block would then be passed to a single column page-segmentation algorithm in order to extract the positions of individual lines. We consider this an acceptable fall-back solution and have developed an interface with these features that is shown in Figure 8.

[**] Most command-line OCR engines (Finereader included) do offer a single-column mode, which does allow some scope for customizing layout-analysis via external tools.

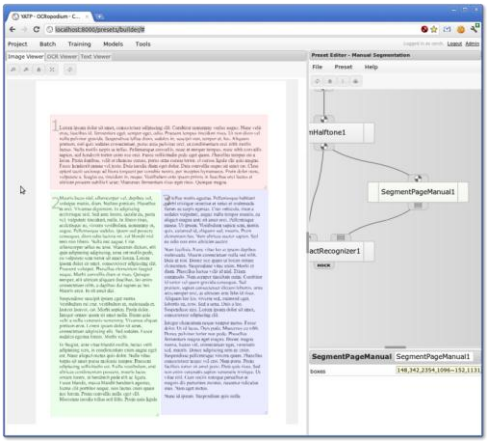



**Figure 8: OWP interface for manual page segmentation**

Both solutions have drawbacks, not least that the problem is completely offloaded to the user. As a compromise between the fully automated and completely manual approaches we have developed a system which attempts to make use of user-supplied hints to derive the correct layout.

After discussion with the research staff responsible for digitising the Stormont Papers, we developed an interface to help address page segmentation problems. Whilst our proof-of-concept is still limited in functionality, it works as an illustration of the idea. The user starts by determining a very general outline that fits all pages they wish to OCR. For example, such an outline might consist of (1) a heading of one line and (2) two additional columns. The algorithm then attempts to determine the best arrangement of page components that fulfils this description. By removing the most basic ambiguities of the target page layout we significantly narrow the scope of the problem, though it remains a non-trivial task to determine the most appropriate complete solution.

This case study provides a good example of how open source technologies can be helpful in enhancing OCR in historical archives. Because of the nature of the material, even a high-quality commercial OCR such as Abbyy Finereader 8.0 CLI will struggle to produce good results without the aid of external pre-processing. Moreover, the flexibility of the environment allows the OCR workflow to be interrupted at any stage, to address specific issues using the staff's knowledge about their collections, as discussed in our example of the page segmentation service.

### 5.2. *Case Study 2: European Holocaust Research Infrastructure*

Our second case study provides a proof-of-concept of how our work could be embedded for enabling scientific workflows in a larger research infrastructure for history.
The European Holocaust Research Infrastructure (EHRI) project[††] aims to create a sustainable Holocaust research infrastructure that will bring together virtual resources from dispersed archives. It will provide open access to Holocaust material such as documents, objects, photos, film and art, and involves 20 partner organisations in 13 countries.

[††] http://www.ehri-project.eu

One of the main challenges of the project is that the dispersed archives of the European Holocaust often do not have the means to digitise their resources to sufficient extent. Similar considerations apply to research projects in the field of the Holocaust, though a number of significant digitisation efforts have been undertaken.

Even when resources are available in digital form, they remain largely inaccessible due to inadequacies of metadata. In this case study we wanted to find out whether we can use the OWP infrastructure to create OCR output that can be used to semantically extract metadata. Semantically enriched library and archive federations have recently become an important part of research in digital libraries (Kruk and McDaniel, 2009), especially as research users generally have more demands on semantics than is generally provided by archival metadata. For instance, place names are often mentioned in the archival descriptions; researchers would like to be able to search for these locations, and place name extraction from the descriptions can help here. By integrating such services within our digitisation workflows, we can provide a good example of how scientific workflows in the humanities can work.

In our proof-of-concept experiments, we demonstrated the principal workflow using off-the-shelf information extraction tools only, and we did not address the broader problems of extracting semantic information from historical texts, which are manifold. For these issues, compare (Packer et al., 2010), as well as (Warner and Clough, 2009), which describes plans for a larger-scale extraction project at the UK National Archives. As for OCR, off-the-shelf commercial software for information extraction has problems with delivering acceptable results for historical material.

Our experiment was undertaken using PDF files of survivor testimonies provided by the Wiener Library[‡‡], an EHRI partner and one of the first Holocaust libraries in the world. The documents were typical fairly low resolution (612x790) grey-scale scans of typed documents, with an index page at the beginning. The PDFs were first converted to a collection of PNG images for use with the web application. Due to the low resolution of the PNGs some attention had to be paid to pre-processing to ensure the most usable OCR output. Prior to binarisation the images were scaled by a factor of four using an anti-aliasing filter to approximate a typical resolution from a 300 DPI scan. This resulted in much cleaner character outlines when binarised, albeit with an inevitable lack of definition. After binarisation, additional filters were applied to de-skew the images and remove edge noise.

The resulting transcript produced by the Tesseract OCR engine was fairly low quality, with around 90% character accuracy. Though this accuracy could well have been improved with further tuning of the pre-processing workflow and/or the use of a different recognition engine, we were mainly interested in how standard information extraction services would react to low-quality textual input. The Tesseract transcript was processed using the standard OpenCalais service provided by ThompsonReuters to extract semantic information (Goddard and Byrne, 2010). Even this standard setup has proven to produce useful results. OpenCalais successfully detected the presence of names in the transcript, even when the OCR was imperfect. For example, it found that “Dmulaltsr Tappe” (Dr. Walter Tappe) was a name, and also marked up several instances of places, such as Berlin and Wilmersdorf. Other incorrectly OCR'd locations such as “Slchsischestraeae” (Schlesische Strasse) were also marked up as places, due to the (correctly OCR'd) preceding phrase “lived in”. Further semantic data marked up by OpenCalais included

[‡‡] http://www.wienerlibrary.co.uk/

job titles ("lawyer", "auditor", "actor") and industry terms ("food"). In several OCR transcripts it detected the topic as "politics". Social tags given included "Berlin", "Geography", and "Geography of Europe".

We consider these initial results to be a qualified success and intend to develop the *OCR-with-semantic-data-extraction* workflow into a full service in the context of the EHRI project. We think we can significantly enhance the research experience on historical archives. At the moment we are concentrating especially on OCRing finding aids from small-scale Holocaust archives.

### 5.3. *Summary of Case Studies*

Each of these case studies shows what becomes possible if we can embed OCR directly in the research workflows at historical archives. The first case study made use of the OWP workflow environment to produce high-quality research-oriented digitisation outputs. In this case, we developed services that allow for the direct linkage of digitisation image and OCR output, ensuring better quality outputs by providing research staff with access to all stages of the interpretation process that translates the digitised image into a textual resource. At the same time, we tried to employ automated processing wherever possible in order not to offload too much work onto busy research staff. Our system of using hints to improve the OCR seems to be a workable intermediate solution.

The second case study demonstrated how open source OCR can work directly within a larger research infrastructure for history. Researchers are commonly interested in accessing archival material using a broader range of "facts" than is provided by standard archival metadata. This case study has shown how this can be addressed by adding semantic information extraction to the OWP workflow, resulting in semantically-enriched outputs that can enhance the research browse experience significantly. This approach has great potential, as we have seen that current information extraction services seem not to depend on a high level of character accuracy in the underlying textual input.

One should add a note of caution here, however. High-quality information extraction is currently limited to a small set of commonly used languages such as English. For EHRI, this is a critical limitation, as much of its novel research material is not written in any of these languages. We believe that, in this case too, open source solutions can help. In particular, we plan to experiment with the GATE environment (Cunningham et al., 2002) for information extraction, supported by community efforts to build up the necessary gazetteers.

## 6. Conclusion and future work

This paper has presented a novel approach to supporting the emerging data-driven humanities. We have described a way of integrating OCR directly into the research and interpretation processes of the humanities by building on open source technologies. The raised by open source technologies often relate to usability, as such technologies generally require a more advanced knowledge of computational environments. We tried to mitigate these issues by developing an easy-to-use workflow environment that we call OWP.

Our evaluation has suggested that the most promising approach for delivering high-quality OCR for historical archives is to combine the commercial engines' robustness and ability to read a wider range of character sets with the flexibility of open

source tools in facilitating customisable pre-processing and layout analysis. OWP serves as a platform for enabling the integration of command-line commercial and open source OCR tools, making hybrid workflows easy to assemble and run in a variety of contexts.

Highly variable material requires a wide range of approaches: character recognition tools vary greatly in their effectiveness depending on the input material, and a one-size-fits-all approach will usually be unsuitable in the context of a typical historical archive. OWP allows users to optimise their OCR approach for particular document sets by offering an easy way of testing and evaluating the effectiveness of different workflow components. We also offer an environment in which the optimised OCR workflows can be linked to specific classes of documents and optimised towards batch-processing in Cloud-based environments.

Finally, we have demonstrated and evaluated in a number of case studies how our environment can enhance existing digitisation processing in small- to medium-scale historical archives. In this paper, we presented two case studies. The first of these discussed and evaluated the general OWP framework against a typical situation in a historical archive, whereas the second showed how OCR can be embedded into wider scientific workflows for historical research on the Holocaust.

Note that, while considerations of cost and financial efficiency formed part of our initial motivations in undertaking the research, via the potential savings to be made by avoiding proprietary software and professional consultancy, they are less significant as a driving factor. While open source software is indeed free, this freedom mainly relates to the liberty with which it may be used rather than to the price attached; as we have ourselves seen, significant effort may be required to adapt it to one's own purposes. Similarly, while our approach avoids costs from professional consultants, it incurs costs through the need for staff to acquire and apply OCR expertise. A full cost-benefit analysis will be possible once our work has progressed further.

The main benefits of our approach relate rather to the *quality* of outcomes than to their cost. As observed above, interpretation is a key element of the digitisation process in humanities research, and this process can thus not be regarded as a neutral or purely technical one. Moreover, archivists themselves have a profound knowledge of the material for which they are responsible, which digitisation activities need to incorporate and exploit. In contrast to the opaque world of proprietary OCR, the flexibility and transparency of open source software permits the knowledge and expertise of such specialised researchers and archivists to be embedded in the process of digitisation, producing results that are of greater scholarly rigour. Moreover, this openness serves to initiate scholarly discussions on the underlying processes involved in transforming analogue research material into its digital surrogates.

Moreover, the close involvement of researchers and archivists in digitisation serves to narrow or close the digitisation "skills gap" in these communities and to furnish them with a fuller appreciation of the problems and opportunities associated with OCR in a scholarly context, an appreciation that can be leveraged in subsequent work. These have many advantages here, as they are open to changes and thus serve to initiate discussions on the underlying processes that are involved in transforming analogue research material into its digital surrogates.

Our immediate next steps will be to enhance our framework and customise it to the needs of the EHRI project. Our initial results are encouraging, and we believe that our approach can have a big impact on research that uses small-scale historical archives that have limited resources to invest in larger digitisation efforts, not to mention in OCR.

These archives dominate the landscape in one of the primary research areas of EHRI: Eastern Europe. We are in the process of developing dedicated workflows to OCR first finding aids and later complete historical collections, should we find additional resources to digitise these. In EHRI, we promised to build a virtual observatory to enable Holocaust research to find and work with disparate Holocaust sources. OCR is an essential component of our plans, and will be the first step towards a deeper semantic access to these archives based on the interests of Holocaust researchers.

## 7. References

AYRIS, P. 2010. The status of digitisation in Europe: extensive summary of the second LIBER-EBLIDA workshop on the digitisation of library materials in Europe. *LIBER Quarterly,* 19**,** 193-226.

BBC. 9/9/2011 2011. Supercomputer predicts revolution. *BBC Online* [Online]. Available from: http://www.bbc.co.uk/news/technology-14841018 2011].

BLANKE, T., BRYANT, M. & HEDGES, M. 2011. Ocropodium – Open source OCR for small-scale historical archives *Journal of Information Science.*

BREUEL, T. 1992. Fast recognition using adaptive subdivisions of transformation space. *In:* IEEE (ed.) *Proceedings of Computer Vision and Pattern Recognition - CVPR '92* Champaign, IL , USA.

BREUEL, T. 2009. Recent progress on the OCRopus OCR system. *Proceedings of the International Workshop on Multilingual OCR.* Barcelona, Spain: ACM.

BROCKS, H., THIEL, U., STEIN, A. & DIRSCH-WEIGAND, A. 2001. Customizable Retrieval Functions Based on User Tasks in the Cultural Heritage Domain. *Lecture notes in computer science.***,** 37-48.

BRYANT, M., BLANKE, T., HEDGES, M. & PALMER, R. 2010. Open Source Historical OCR: The OCRopodium Project Research and Advanced Technology for Digital Libraries. *In:* LALMAS, M., JOSE, J., RAUBER, A., SEBASTIANI, F. & FROMMHOLZ, I. (eds.). Springer Berlin / Heidelberg.

CHEN, X., WILLS, G., GILBERT, L. & BACIGALUPO, D. 2010. Using Cloud for Research: A Technical Review.

COX, P. T. & GAUVIN, S. 2011. Controlled dataflow visual programming languages. *Proceedings of the 2011 Visual Information Communication - International Symposium.* Hong Kong, China: ACM.

CRANE, G. 2002. Cultural Heritage Digital Libraries: Needs and Components. *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries.* Springer-Verlag.

CUNNINGHAM, H., MAYNARD, D., BONTCHEVA, K. & TABLAN, V. 2002. GATE: an architecture for development of robust HLT applications. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.* Philadelphia, Pennsylvania: Association for Computational Linguistics.

DUGUID, P. 2007. Inheritance and loss? A brief survey of Google Books. *First Monday,* 12.

DUNNING, A., ANDERSON, S., POLFREMAN, M., ALBUQUERQUE, V. & HEDGES, M. 2007. The Stormont Papers: from Partition to Direct Rule - 50 years of Northern Ireland Parliamentary papers online. 2007 ed.

GODDARD, L. & BYRNE, G. 2010. Linked Data tools: Semantic Web for the masses. *First Monday,* 15**,** 1.

HAIGH, S. November 1996. Optical character recognition (ocr) as a digitization technology. *In:* CANADA, N. L. O. (ed.) *Technical report, Information Technology Services.*

JANKOWSKI, N. W. 2009. *E-Research: Transformation in Scholarly Practice,* New York, Routledge.

KRUK, S. R. & MCDANIEL, B. 2009. *Semantic Digital Libraries,* Berlin, Heidelberg, Springer Berlin Heidelberg.

MENDES, P., JAKOB, M., GARCÍA-SILVA, A. & BIZER, C. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. *In the Proceedings of the 7th International Conference on Semantic Systems (I-Semantics).*

MORI, S., NISHIDA, H. & YAMADA, H. 1999. *Optical character recognition*, John Wiley \\& Sons, Inc.

MORI, S., SUEN, C. Y. & YAMAMOTO, K. 1992. Historical review of OCR research and development. *Proceedings of the IEEE,* 80**,** 1029-1058.

NEUDECKER, C. & TZADOK, A. 2010. User Collaboration for Improving Access to Historical Texts. *LIBER Quarterly,* 20**,** 119-128.

OINN, T., GREENWOOD, M., ADDIS, M., ALPDEMIR, M. N., FERRIS, J., GLOVER, K., GOBLE, C., GODERIS, A., HULL, D., MARVIN, D., LI, P., LORD, P., POCOCK, M. R., SENGER, M., STEVENS, R., WIPAT, A. & WROE, C. 2006. Taverna: lessons in creating a workflow environment for the life sciences: Research Articles. *Concurr. Comput. : Pract. Exper.,* 18**,** 1067-1100.

PACKER, T. L., LUTES, J. F., STEWART, A. P., EMBLEY, D. W., RINGGER, E. K., SEPPI, K. D. & JENSEN, L. S. 2010. Extracting person names from diverse and noisy OCR text. *Proceedings of the fourth workshop on Analytics for noisy unstructured text data.* Toronto, ON, Canada: ACM.

PAYETTE, S. & LAGOZE, C. 1998. Flexible and Extensible Digital Object and Repository Architecture (FEDORA)

Research and Advanced Technology for Digital Libraries. Springer Berlin / Heidelberg.

PLOEGER, L. 2009. In Brief: IMPACT. *D-LIB Magazine*, 15.

RICE, S. V., NAGY, G. L. & NARTKER, T. A. 1999. *Optical Character Recognition: An Illustrated Guide to the Frontier*, Kluwer Academic Publishers.

SMITH, J. W. & MERALI, Z. 1985. *Optical character recognition : the technology and its application in information units and libraries*, London, British Library.

SMITH, R. An Overview of the Tesseract OCR Engine. Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, 23-26 Sept. 2007 2007. 629-633.

THOMPSON, C. July 1997. OCR/ICR Accuracy and Acceptance- What does it mean? . *Inform magazine*.

VAMVAKAS, G., GATOS, B., STAMATOPOULOS, N. & PERANTONIS, S. J. A Complete Optical Character Recognition Methodology for Historical Documents. Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on, 2008. 525-532.

VOLK, M., FURRER, L. & SENNRICH, R. 2011. Strategies for reducing and correcting OCR error. *In:* SPORLEDER, C., BOSCH, A. V. D. & ZERVANOU, K. (eds.) *Language Technology for Cultural Heritage.* Berlin: Springer.

WARNER, A. & CLOUGH, P. 2009. A Proposal for Space Exploration at The National Archives. 2011. Available: http://ir.shef.ac.uk/cloughie/papers/York2009.pdf.

## About the authors

TOBIAS BLANKE[§§], Email: tobias.blanke@kcl.ac.uk

Tobias Blanke is a Senior Lecturer in the Centre for e-Research at King's College London. He is a director of DARIAH, a European research infrastructure for arts, humanities and cultural heritage data, and leads the joint research work for EHRI, a pan-European consortium to build a European Holocaust Research Infrastructure. His academic background is in philosophy and computer science.

MICHAEL BRYANT , Email: michael.bryant@kcl.ac.uk

Michael Bryant is a Research Associate in the Centre for e-Research at King's College London. He has been leading the development work for OCRopodium and is currently working on EHRI as a development lead.

MARK HEDGES, Email: mark.hedges@kcl.ac.uk

Mark Hedges is a Senior Lecturer at the Centre for e-Research at King's College London. He directs the Centre's research strategy, and is leader of a number of research projects in the fields of e-research and digital libraries. His academic background is in mathematics and philosophy – he has a Ph.D. in mathematics – and, more recently, in Byzantine studies.

[§§] Corresponding Author